An introduction to nomography: Garrigues' nomogram for the computation of Easter

Denis Roegel

Abstract

This article analyzes a calendrical nomogram for the determination of the date of (Julian or Gregorian) Easter, and shows how it can be reproduced with METAPOST.

1 Introduction

The field of nomography is ancient, and is related to slide rules. The object of nomography is to study the graphical representation of equations with n unknowns, in order to construct graphical tables representing mathematical laws of which these equations are the analytical expression. These tables are called "nomograms" and can be used to obtain one of the n values given the values of the $n-1$ other unknowns.

The art of nomography was developed extensively by Maurice d'Ocagne (1862–1938), from 1884 onwards. In his 1921 treatise on the subject, he mentioned a nomogram for the calendar, as well as the unpublished work of André Crépin on one for finding Easter (d'Ocagne, 1921, p. 468–470). Then, in 1939, Damien Garrigues published an article with a nomogram for finding the date of Easter in the Julian and Gregorian calendars (Garrigues, 1939). Garrigues did not refer to Crépin, and may have constructed his nomogram independently.

Our article explores this particular example and shows how this nomogram can be reproduced using METAPOST.[1] We will first analyze the structure of Garrigues' nomogram, and we will need to review some basic information on the calendar. Once we have a good grasp of the principles underlying Garrigues' nomogram, we will examine how to tackle its graphical challenges with METAPOST.

2 Easter in the Christian calendar

Easter is a Christian feast commemorating the resurrection of Christ and has been celebrated since the first centuries of our era. As time went by, it was decided to set the date of Easter on the Sunday immediately following the first full moon of Spring. For practical reasons, Spring is considered to begin on March 21st, and the full moons are based on simplified tables, not on astronomical observations. This rule applies both to the Julian calendar (before the Gregorian reform which took place in 1582) and to the Gregorian calendar. However, the tables for the paschal lunar phases were made more accurate in 1582, and the average year was made slightly shorter, so that this actually made the computation of Easter more complex.

We will not enter into the details of the history of the Christian calendar, nor of the many algorithms for the computation of Easter, but we will summarize — without proof — the basic procedure for the computation of Easter. More detailed information on calendrical calculations can be found in the standard (Dershowitz and Reingold, 2008), but information on Easter algorithms is scattered in multiple other sources. Besides (Gauss, 1973), one can consult (Knuth, 1997) for a simple (but exact) algorithm, and (Bien, 2004) for a comparison between a few Easter algorithms.

2.1 Julian calendar

In the Julian calendar, the date of Easter repeats itself after exactly 532 years. The computation is based on a lunar cycle of 19 years (the phase of the moon was supposed to be again the same after 19 years) and on a (solar) calendar cycle of 28 years (the years repeat after 28 years, since every fourth year is a leap year, since common years would repeat after seven years, and $28 = 4 \times 7$). We then merely have an Easter cycle of $532 = 19 \times 28$ years.

In this calendar, the position of a year in the 19 year lunar cycle is given by its Golden Number G:

$$Y \leftarrow \text{year} \quad (1)$$

$$G \leftarrow (Y \bmod 19) + 1 \quad (2)$$

Using the Golden Number, the (Julian) epact E_J of the year can be computed: the epact (in its modern sense) is the age of the moon on January 1st, minus one (Roegel, 2004). Since the moon phases shift by about 11 days every year, the epact consequently increases by about 11 units every year. It can be obtained from the Golden Number as follows:

$$E_J \leftarrow (11G - 3) \bmod 30 \quad (3)$$

And the value of the epact then determines the date of paschal full moon.

Sundays are determined by what is called the "dominical letter". All the days of a common year can be labeled by a letter from A to G, starting with A on January 1st, B on January 2nd, etc., G on January 7th, A again on January 8th, etc., reaching C on February 28, and D on March 1st (February 29 is considered to be without a letter). The "dominical letter" is then merely the letter associated to the Sundays of a year. When the year is a leap year, there are of course two dominical letters, one for January and February, and one for the other ten

[1] The complete METAPOST code is available on CTAN under the name `garrigues.mp`.

months, because the layout of the letters is defined for common years.

The date of Easter is obtained by combining the epact and the dominical letter.

2.2 Gregorian calendar

In the Gregorian calendar, the phases of the moon do no longer follow a 19 year cycle. The new cycle is more complex, as a consequence of a more accurate modeling of the mean motion of the moon, and because of the shorter mean solar year. The computation can still be based on the Golden Number and the (Julian) epact, but the epact is corrected as follows. We first define the secular part S of the year, then a correction M:

$$S \leftarrow \left\lfloor \frac{Y}{100} \right\rfloor \tag{4}$$

$$M \leftarrow \left(15 + S - \left\lfloor \frac{S}{4} \right\rfloor - \left\lfloor \frac{8S+13}{25} \right\rfloor\right) \bmod 30 \tag{5}$$

It was Gauss who introduced M in this form in 1816 (Gauss, 1800; Gauss, 1816).

What we call the "mean Gregorian epact" $\overline{E_G}$ is defined as follows:

$$\overline{E_G} \leftarrow (E_J - (M - 15)) \bmod 30 \tag{6}$$

The previous correction to E_J can also be used for the Julian calendar, by taking $M = 15$. In that case, $\overline{E_G} = E_J$.

The real (or corrected) Gregorian epact E_G, instead, is given by:

$$E_G \leftarrow \begin{cases} \overline{E_G} + 1 & \text{if } (\overline{E_G} = 25 \text{ and } G > 11) \\ & \quad \text{or } (\overline{E_G} = 24) \\ \overline{E_G} & \text{otherwise} \end{cases} \tag{7}$$

This value of the epact can be used to obtain a full moon in March. N_1 is the day in March for a full moon, but it may be another full moon than the paschal full moon (full moon on which the definition of Easter is based):

$$N_1 \leftarrow 44 - E_G \tag{8}$$

The real paschal full moon in March is:

$$N_2 \leftarrow \begin{cases} N_1 + 30 & \text{if } N_1 < 21 \\ N_1 & \text{otherwise} \end{cases} \tag{9}$$

Garrigues' nomogram computes the paschal full moon without the corrections for E_G, and obtains a date of Easter. Ignoring the corrections on the epact produces certain wrong epacts, but only some of these wrong epacts cause an incorrect date of Easter. The dates are incorrect in only rare circumstances, which are listed in the nomogram (1954, 2049, 2106, etc.) and which will be analyzed later in this article.

3 The structure of Garrigues' nomogram

3.1 An example

Garrigues' article shows the use of the nomogram for the year 1939, the year the article was published. Using the nomogram is straightforward. The year is first divided in its century number (called "partie séculaire", or secular part in French) and the last two digits of the year (merely called "Année", that is, year in French). Each of these parts is looked up in columns I and III (figure 1) and the centers of the two circles containing the values sought are connected by a dashed line. This line falls on a point in column II, and this point is in turn connected to the first point at the top of column IV. This is the Golden Number associated to 1939.

The secular part is reused in column VI, and joining it with the Golden Number just found, a new point is obtained in column V. This point is connected to the point labeled 10 in column VII, and this is the value of the (mean) Gregorian epact.

Now, using again the secular part in the right part of column VIII and the last two digits of the year in column X, we obtain a point labeled "A" in column IX. This point is connected to point "A" in column XI. Finally, the intersection of the lines connecting point 10 of column VII and point B on one hand, and point "A" of column XI and point C on the other hand, falls in the slot corresponding to April 9, which is the date of Easter in 1939.

Before attempting to reproduce the nomogram, we will first try to analyze its construction. This will provide us with enough insight and will lead seamlessly to the METAPOST code.

As we have just seen, Garrigues' nomogram is made of several parts, which are all fairly regular. The areas were numbered by Garrigues in Roman numerals I, II, III, ..., XVI, but in this article we will only consider the first eleven areas, the only ones which are concerned with the calculation of Easter. We will analyze each of these areas in sequence.

It is important to understand the geometry of the nomogram, because the geometry represents the relationship between the variables.

3.2 Basic features of the nomogram

The basic features of Garrigues' nomogram are the following:

- some lines or sequences of points are annotated using various functions: a set of points 1, 2, ..., i are distributed linearly and annotated with $f(i)$; examples are given in figure 2;
- additions are obtained by drawing a line: the addition is on the index values, that is, on po-

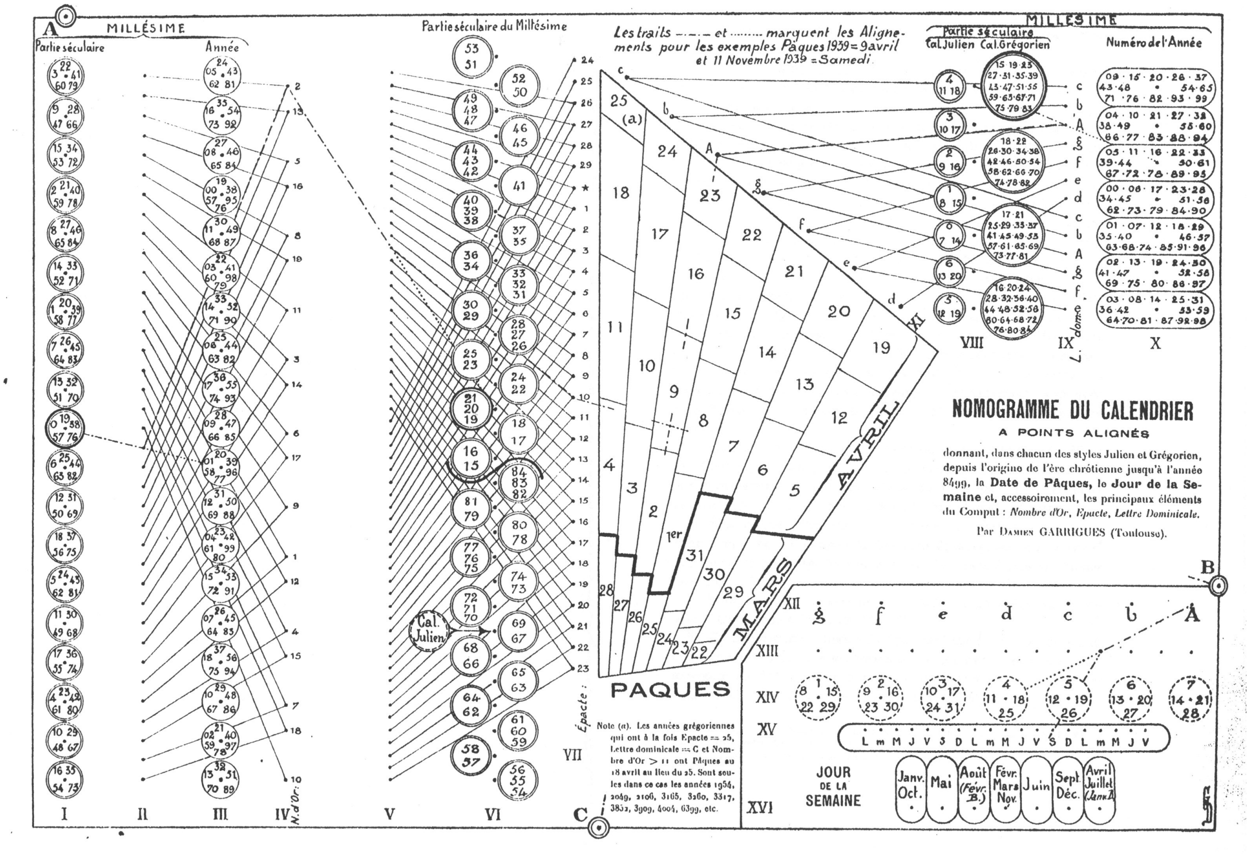

Figure 1: Garrigues' original nomogram (excerpt from (Garrigues, 1939)).

sitions; this scheme is used here three times; in each case, from two values among n values, we obtain $2n-1$ combined values.

- for columns I–III, $n = 19$;
- for columns IV–VI, $n = 30$;
- for columns VIII–X, $n = 7$.

We first consider the scheme represented on the left of figure 2. For $i = 0, 1, \ldots$, let $c(p_i) = (0, i)$, $c(q_i) = (2, i)$ and $c(r_i) = (1, i/2)$ be the coordinates of points p_i, q_i and r_i, and let $v(p_i)$, $v(q_j)$ and $v(r_k)$ be the values associated to p_i, q_j and r_k. We have of course $v(p_i) = i$, $v(q_j) = j$ and $v(r_k) = k$. Let $v'(p)$ be the value associated to the point at coordinates p, then $v'((0,i)) = i$, $v'((2,i)) = i$, and $v'((1,i/2)) = i$. Finally, $v'(c(p_i)) = i$, $v'(c(q_j)) = j$, and $v'\left(\frac{c(p_i)+c(q_j)}{2}\right) = v'((1,(i+j)/2)) = i+j$. The example shows how we obtain 5 by adding 2 and 3.

On the right of figure 2, instead, we do not add 2 and 3, but we obtain the position 5 from positions 2 and 3. 2, 3 and 5 are index values, not the values sought themselves. So, the scheme on the right can be used to compute z_{i+j} from x_i and y_j, but the value of z_{i+j} need not be the sum of x_i and y_j. The first case is of course a special case of the second one, where $x_i = i$, $y_j = j$ and $z_k = k$.

This scheme is used three times in Garrigues' nomogram, with x_0, y_0 and z_0 at the bottom in the three cases. In columns I–III (see figure 3), x_i is the sequence of Golden Numbers 4, 12, 1, 9, 17, we can take $y_i = x_i$ (or any other shifted sequence x_{i+s}), and z_i is the sequence 18, 7, 15, 4, 12, 1, 9, etc.

In columns IV–VI (figure 5), $x_i = E_J(i)$ (Julian epact), $y_i = 15 - M(i)$ and $z_i = \overline{E_G}(i)$ (mean Gregorian epact).

In columns VIII–X (figure 7), x_i, y_i and z_i are values associated to dominical letters.

- some values are rearranged: data can be transferred from one line to another, using a mapping; $f(i) = g(h(i))$ where $f(i)$ is the function on the first line, $g(j)$ is the function on the second line, and $j = h(i)$ is the mapping from one

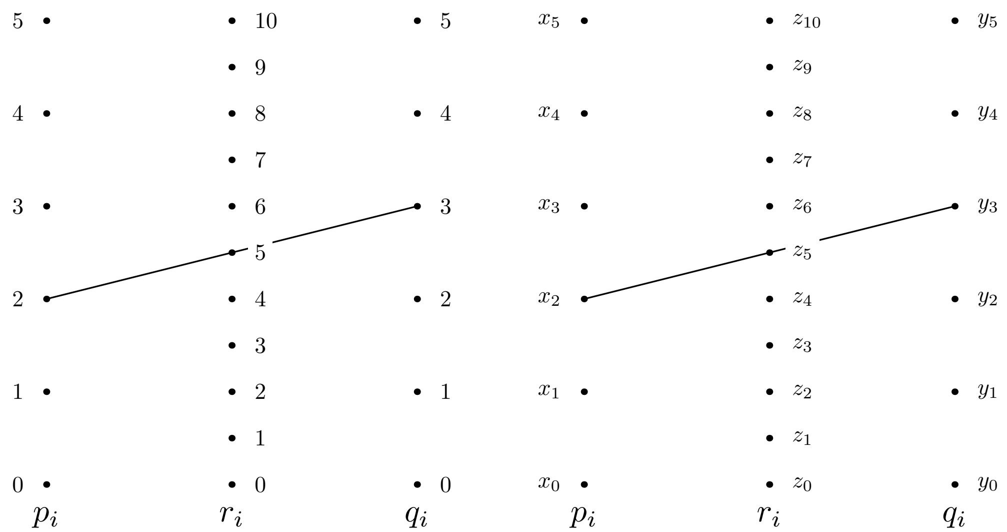

Figure 2: Basic addition on a nomogram: direct addition (left), and addition on indices (right). This scheme is used three times by Garrigues. The left part is a special case of the right one with $x_i = i$, $y_i = i$, and $z_i = i$.

line to the other; see for example figure 4.

- certain values are obtained as intersections in a 2-dimensional grid: from two lines indexed by i and j, a grid can be constructed from the values of $f(i, j)$. See for instance figure 10.

3.3 Description of the components of Garrigues' nomogram

The columns of the nomogram will be described in the following order, not strictly from left to right.

Columns I–III (figure 3) The purpose of the first three columns is to obtain the Golden Number G corresponding to a given year. The year is identified by its secular part S and by its last two digits A. The arrangement of columns I and III is a consequence of the arrangement of column II. We therefore first need to understand column II and then we can proceed with columns I and III.

Column II: The points in this column represent values of the Golden Number G, from top to bottom: 2, 13, 5, 16, 8, 19, 11, 3, 14, 6, 17, 9, 1, 12, 4, 15, 7, 18, 10, and then again 2, 13, ..., until 18 (the values are shown in column IV). This is the order of the Golden Numbers if they are rearranged by the corresponding values of the Julian epacts E_J which are 19, 20, (21), 22, 23, (24), 25, 26, (27), 28, (29), 0, 1, (2), 3, 4, (5), 6, (7), 8, 9, (10), 11, 12, (13), 14, 15, (16), 17, (18). (Values between parentheses do not occur as Julian epacts, hence the gaps in column IV.) So, $G = 2$ corresponds to Julian epact 19, $G = 13$ corresponds to Julian epact 20, and so on. Let $c_2[i]$ be the i-th Golden Number value (from the bottom) in column II: we have $c_2[1] = 18$, $c_2[2] = 7$, etc. It is easy to see that $c_2[i] = 1 + ((9 + 8i) \bmod 19)$. We can also write $c_2[20-i] = (5+11(i-1)) \bmod 19 = (13+11i) \bmod 19$, which shows that the Golden Numbers increase by 11 (mod 19) from top to bottom.

Column I: The first column is related to the secular parts S of the years, that is, the digits left when removing the last two digits of the year. 2008, for instance, has 20 for its secular part S. The secular parts are arranged by their remainder by 19 and there are therefore 19 circles with values inside. However, the circles are not ordered in the usual order, that is, remainder 1 does not follow remainder 0, remainder 2 does not follow remainder 1, etc. Instead, the secular parts are ordered according to their contribution to the Golden Number in column II. So, $S = 9$ follows $S = 3$ because $900 = 47 \times 19 + 7$, and $300 = 15 \times 19 + 15$, and 7 follows 15 in column II, and so on.

We refer to these circles by the smallest values found inside, namely 3, 9, 15, 2, 8, ..., 16. Two consecutive values differ by 6 (mod 19), because adding 6 to a secu-

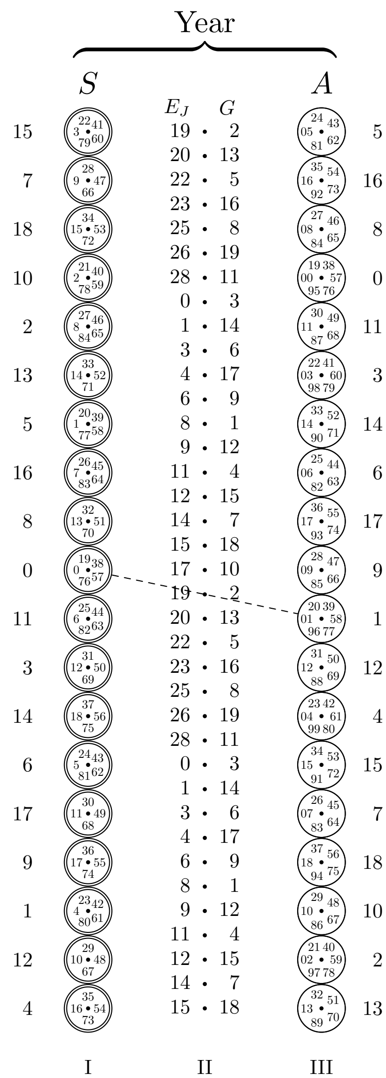

Figure 3: Finding the Golden Number in column II, using the components of the year. The Golden Numbers G are given in the order of the Julian epacts E_J, but without gaps. We have added the values of $(100S) \bmod 19$ (left of column I) and $A \bmod 19$ (right of column III).

lar part is equivalent to adding 11 to the Golden Number ($6 \times 100 \equiv 11 \pmod{19}$), and we have seen that the Golden Numbers in column II increase by 11 (mod 19) from one point to the next point below.

Moreover, values such as 3, 22, 41, etc., are located in the same circles since they are equal modulo 19. If we call $c_1[i]$ the index value on the i-th point in column I (with point 1 at the bottom), we can see that $c_1[i] = (22 - 6i) \bmod 19$. For instance, $c_1[17] = (22-6\times 17) \bmod 19 = 15$. When the secular part is S, it goes on the point $point_S(S) = 1 + ((3S + 9) \bmod 19)$. For instance, if $S = 19$, $1 + ((3 \times 19 + 9) \bmod 19) = 10$, so 19 is located on the 10th point. We can check that $\forall S < 19 : c_1[point_S(S)] = S$.

Column III: The last two digits of the year are also positioned in relation with the second column, for the same reason as for column I; we can notice that the years appear in the order (from top to bottom) 5, 16, 8, 19, 11, etc., exactly like the order of the values in column II; that is, $c_3[i] = (5 - 11i) \bmod 19$, where $c_3[i]$ is the smallest value in a circle in column III. For instance, on the first point, $c_3[1] = (5 - 11) \bmod 19 = 13$. On the second point $c_3[2] = 2$, etc. The point corresponding to the last two digits A of the year is $point_A(A) = 1 + ((15 + 12A) \bmod 19)$ and one can check that $\forall A < 19 : c_3[point_A(A)] = A$.

Linking columns I and III: If the centers of the circles in the first column are at coordinates $(0, i-1)$ where i is the point number (in the enumeration given above), and if the centers of the circles in the third column are at coordinates $(2, j)$ with $j = 0$ to 18 ($j = 0$ at the bottom), then the points in the second column are located at coordinates $(1, k/2)$ where $k = 0, 1, \ldots, 36$. This is the scheme shown in figure 2.

We can now check that linking the secular part and the last digits of the year indeed gives the Golden Number. Figure 3 shows columns I to III, and, for every value of S and A, we have added the value of $(100S) \bmod 19$ (left of column I) and of $A \bmod 19$ (right of column III). We have also added the values of the Golden Number G in column II. We are in the conditions of figure 2, where $x_i = (4 + 8i) \bmod 19$, $y_i = (13 + 8i) \bmod 19$, and $z_i = (18 + 8i) \bmod 19$. The case $i = j = 0$ corresponds (for instance) to the year 1613, for which $G = (1613 \bmod 19)+1 = 18$. Therefore, the triplet (x_0, y_0, z_0) is indeed a correct one. What we need to prove is that any three aligned points make a correct triplet. A correct triplet is of the form (x_i, y_j, z_{i+j}) (figure 2). This can be proved by induction. If we assume that the triplet (x_i, y_i, z_{i+j}) is a correct triplet, then, since $(x_{i+1}-x_i) \bmod 19 = (y_{i+1}-y_i) \bmod 19 =$

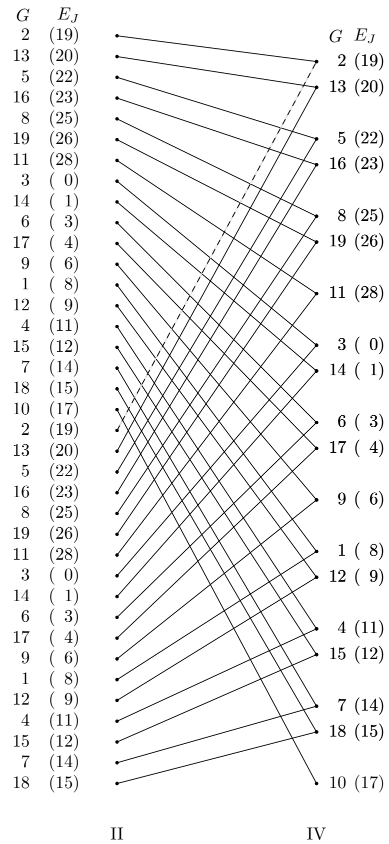

Figure 4: Rearranging the Golden Numbers (column II) according to the Julian epacts (column IV), but with gaps for the missing epacts.

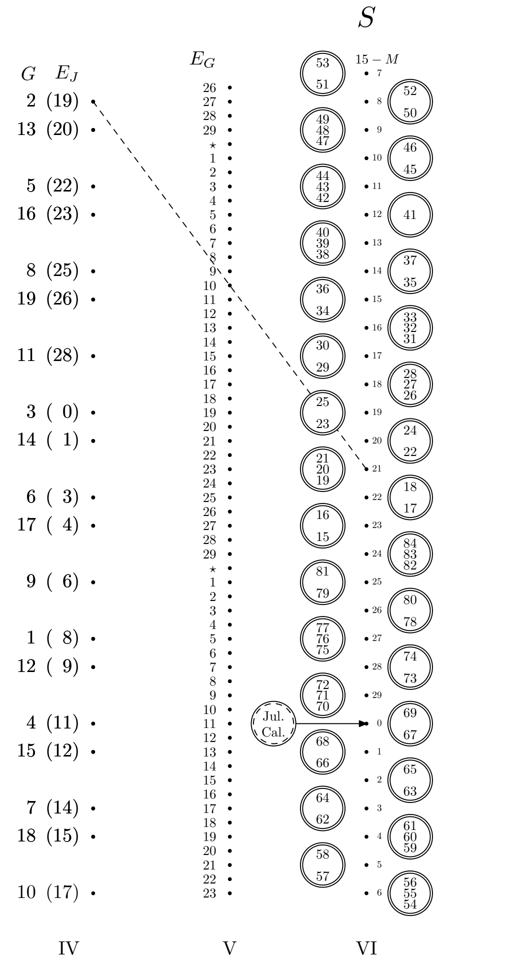

Figure 5: Finding the mean Gregorian epact (column V) using the Julian epact (column IV) and the value of M (column VI). If we add E_J to $15-M$, we obtain the mean Gregorian epact $\overline{E_G}$. For $i=0$, $E_J=17$, $M=9$, $15-M=6$ and $\overline{E_G}=E_J-(M-15)=23$. When M increases, $\overline{E_G}$ decreases. When E_J decreases, $\overline{E_G}$ decreases too. The left column shows $x_i=(17-i) \bmod 30$, the right column shows $y_i=(6-i) \bmod 30$ and the middle column shows $z_i=(23-i) \bmod 30$.

$(z_{i+1}-z_i) \bmod 19 = 8$, it follows that $(x_{i+1}, y_i, z_{i+j+1})$ and $(x_i, y_{i+1}, z_{i+j+1})$ are also correct ones, because z_k increased by exactly as much as either the secular part or the last digits of the year contributed to the Golden Number, and therefore z_k must still be the Golden Number.

Garrigues could have designed the columns I–III more naturally, by putting the Golden Numbers in their natural order, but he chose to put them in the order of the Julian epacts without showing the values of those.

Columns IV–VI: (figure 5) The purpose of these three columns is to compute the mean Gregorian epact $\overline{E_G}$ from the Julian epact E_J.

Column V: This column is found halfway between columns IV and VI. Column IV represents the Julian epact E_J and column VI corresponds to the correction $15-M$, with $M=(15+S-\lfloor\frac{S}{4}\rfloor-\lfloor\frac{8S+13}{25}\rfloor) \bmod 30$, where $\lfloor\ldots\rfloor$ is the integer part. M is Gauss' correction (Gauss, 1816). The mean Gregorian epact $\overline{E_G}$ is given by $\overline{E_G}=(E_J-(S-\lfloor S/4\rfloor-\lfloor(8S+13)/25\rfloor)) \bmod 30 = (E_J+(15-M)) \bmod 30$. We call $\overline{E_G}$ the "mean Gregorian epact", because it is the epact considered without the corrections for epacts 24 and 25.

The real Gregorian epact E_G sometimes differs from the value of the mean Gregorian epact $\overline{E_G}$ which is obtained from the nomogram. $E_G=\overline{E_G}+1$ if ($\overline{E_G}=25$ and $G>11$) or ($\overline{E_G}=24$). This shifts the paschal full moon one day earlier. We will come back to these exceptions when examining the Easter area between columns VII and XI.

Column V can therefore be seen as the sum of columns IV and VI, and will be the mean Gregorian epact $\overline{E_G}$. This column has 58 points, that is, as many as there are combinations between columns IV and VI. The values of the epact are shown in column VII. Epact 0 or 30 is usually written '$\star$'.

Column IV: This column only serves for an addition with the value represented in column VI. There are 30 points in column IV if one includes the gaps. As mentioned above, the point numbers correspond to the Julian epacts on a 1–30 scale (hence the gaps). Each point corresponds to a value of the Julian epact, but the points are labeled with the Golden Number, since there is an exact correspondence between them. When the Golden Number is G, the Julian epact E_J is $(11G-3) \bmod 30$ and E_J is on point $1+((17+29E_J) \bmod 30)$ hence on point $1+((20+19G) \bmod 30)$. $G=1$ corresponds to point 10, $G=2$ to point 29, $G=3$ to point 18, $G=4$ to point 7, etc. The first point on the top of column IV corresponds to $E_J=19$ (for $G=2$). The empty slot above it would correspond to Julian epact 18, but such a value does not exist in the Julian calendar. The second point from the top corresponds to $E_J=20$ (for $G=13$), the second empty slot to $E_J=21$ (which does not exist), the next point is $E_J=22$ (for $G=5$), and so on, until Julian epact 17 (for $G=10$) at the bottom.

So, column IV shows the Golden Number, but at positions corresponding to the Julian epacts.

Column VI: The secular parts S of the year, between 15 and 84, are positioned according to the values of $M=(15+S-\lfloor S/4\rfloor-\lfloor(8S+13)/25\rfloor) \bmod 30$; 30 points are in this column. The first point at the top corresponds to $M=8$, the second point to $M=7$, etc., until $M=9$ at the bottom (for $S=54$, 55, and 56). If S corresponds to a value M, it is put on the $(1+(M+21) \bmod 30)$-th point from the bottom. 30 different circles are put along that column, left and right of it, to save space. It was certainly this column which led Garrigues to stop the secular parts at 84, because $S=85$ would have had to be added to the circle with $S=15$ and $S=16$, breaking the evenness of the distribution of S. Nevertheless, the nomogram could easily be extended if necessary.

This column is used together with the Golden Number G (column IV) to obtain the epact (column V). The Julian calendar corresponds to $M=15$ which goes with $S=67$ and 69 in the Gregorian calendar.

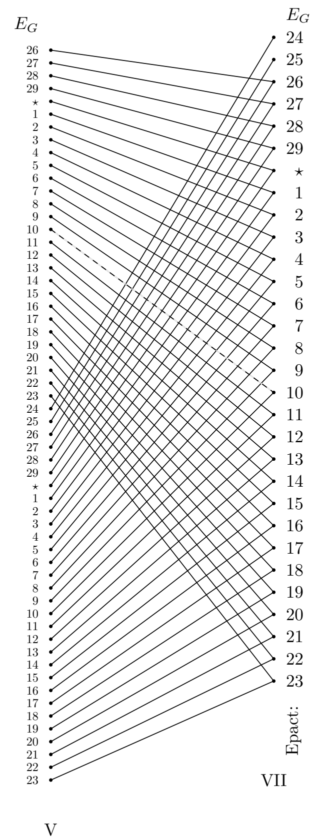

Figure 6: Rearranging the mean Gregorian epacts uniquely in column VII.

Column VII: This column is like column V, but it will be used for the right-hand side of the drawing. The values do not start at the bottom, but this doesn't matter, as we have some freedom in the positioning of the points.

Columns VIII–X: (figure 7) The purpose of these three columns is to obtain the dominical letter of the year, using the secular part S and the last two digits A of the year.

Column IX: This column gives the series of dominical letters DL from bottom to top,

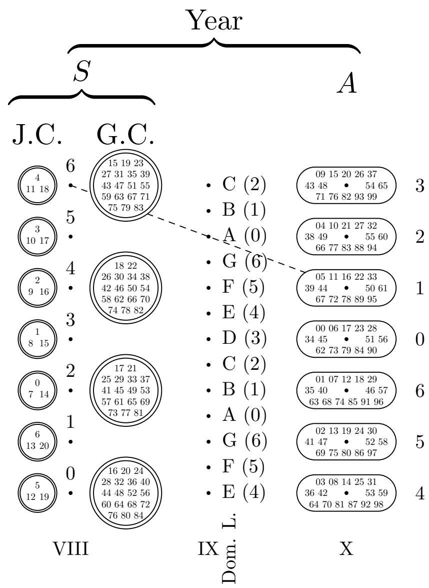

Figure 7: Finding the dominical letter (column IX) of a year, using its components S and A. The 1st of March 1900 was a Thursday (d.l. = G). Consequently, the 1st of March 2000 was a Wednesday (and d.l. = A). The 1st of March 2100 will be a Monday, and so on. If the vertical scale in column VIII is such that the days of the week go up from top to bottom, the centuries will be arranged as on the figure and the dominical letters must go in the opposite direction. In the Julian calendar, there is always a gap of 1 between two centuries. This figure shows the second dominical letters of the years $100S$ over the points of column VIII, with the convention $A \to 0$, ..., $G \to 6$. The first line corresponds to 2008 (for instance) and $z_0 = x_0 + y_0$.

starting with E. Each letter is associated with a number: $A \to 0$, ..., $G \to 6$.

Column VIII: For a year $Y = 100S + A$, column VIII corresponds to the day of the week for the 1st of March of year $100S$, and hence also to the second dominical letter of year $100S$ (the first dominical letter is for January and February). The case where March 1st is a Wednesday is put at the bottom and corresponds to the dominical letter A (because the letter associated with March 1st is D, and the previous Sunday is on the letter A). This is the case for 1600 and 2000, for instance. In the Julian calendar, since 100 years make up 36525 days ($\equiv 6 \bmod 7$), advancing 100 years means going one day backwards in the week and one letter forward in the dominical letters. This is shown in column VIII on the left when we go up when the century increases.

In the Gregorian calendar, there are either 36525 days or 36524 days in a century. Hence, we go up by two days, except when going from $S = 19$ to 20, from $S = 23$ to 24, etc.

Column X: The values on the right of column X show the second dominical letters in the 21st century, with the same conventions as in column IX. 2000, for instance, had dominical letter A, and the last digits 0 of 2000 fall at position 0 (figure 7), 0 being associated with the dominical letter A.

Since the value associated at the bottom of column VIII is 0, and since the first value in column IX is 4, the years such as 2008 (with DL equal to E) are also put at the bottom of column X.

We can see how the other values are laid out: 365 days are a multiple of 7 plus 1. So every time we have a new year, we add one to the day of the week, except when we pass from a year like $4n - 1$ to $4n$ (for instance from 2007 to 2008). The year A goes on point $1 + (3 - A - \lfloor A/4 \rfloor) \bmod 7$ (counted from 1 at the bottom).

Linking columns VIII and X: Column VIII corresponds to the day of the week of the first March of the first year with secular number S, and column X corresponds to the shift introduced by the year within the 21st century. Adding the two gives the day of the week for the 1st of March of the year considered, because 2000 is at the bottom of column VIII, and hence gives the second dominical letter of the year. The values of column IX follow.

Column XI: This column reproduces the values of column IX, but avoiding the duplication.

Easter area: This is a table giving Easter using the mean Gregorian epact $\overline{E_G}$ and the dominical letter DL. Points B and C are used to draw lines towards the epact and dominical letter values, and the intersections fall in a slot. There are 35 possible days for Easter and therefore 35 slots in this area. Basically, the epact gives us the day of the pascal full moon, and the dominical letter gives us the day of the week of that full moon. The two together give the date of Easter. There are five Easter dates corresponding to each dominical letter.

This table must take the epact exceptions into account. As we have seen earlier, the value

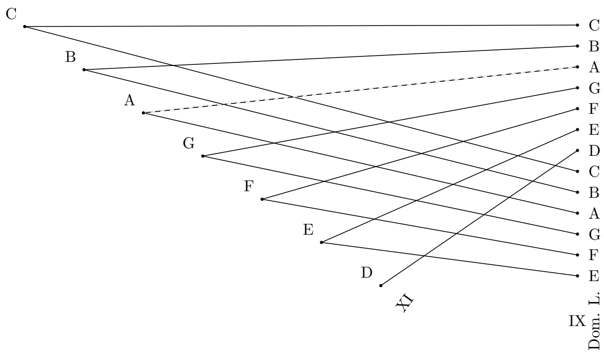

Figure 8: Rearranging the dominical letters for use in the Easter area (column XI).

of the mean Gregorian epact is subject to a correction in certain cases. If there are no corrections, a (mean) epact value of 24 corresponds to a paschal full moon on April 19, and a (mean) epact value of 25 corresponds to a paschal full moon on April 18. The corrections have as an effect to always shift the epact 24 full moon to April 18, and to move the epact 25 full moon to April 17 only when the Golden Number is greater than 11. This means that the cases $\overline{E_G} = 24$ and $(\overline{E_G} = 25) \wedge (G \leq 11)$ correspond to the same paschal full moon (April 18), and this is what is shown in figure 10. The only exception in the Easter area table is therefore the case $(\overline{E_G} = 25) \wedge (G > 11)$. In this case, the new paschal full moon is April 17, and this will only cause the date of Easter to move if April 18 happened to be a Sunday, hence if the dominical letter was C.

So, the table would give the date of Easter April 25 when the mean Gregorian epact is 25, the dominical letter is C, and the Golden Number is greater than 11. Between 1583 and 10000, this occurs only in 1954, 2049, 2106, 3165, 3260, 3317, 3852, 3909, 4004, 6399, 6551, 7086, 7143, 7238, 8202, 8297, 8354, 8449, and 9041. In these cases, the date of Easter should therefore be April 18 and not April 25 and this is what Garrigues indicated in a footnote.

4 Reproducing the nomogram with METAPOST

Reproducing Garrigues' nomogram in METAPOST is easy, once we have a good understanding of its structure. The complete reconstructed nomogram is shown in figure 9. We will in turn consider the positions of the points, the connections, the labels, and the Easter grid (between columns VII and XI).

4.1 METAPOST

METAPOST is the graphical programming language accompanying TeX. Graphics are expressed as programs where various points, lines and labels are defined. We will not describe the language here, and we refer the reader to the main references (Goossens, Mittelbach, Rahtz, Roegel, and Voss, 2008; Hobby, 2008). However, in the sequel, we will explain some of the interesting or particular constructions used in our code.

4.2 LaTeX labels with the `latexmp` package

TeX labels are usually included in METAPOST using the `btex ... etex` construction, but this is a very inefficient solution, especially when labels are parameterized. A much better solution is to use the `latexmp` package which provides a macro `textext` taking a string representing some LaTeX code. This is what we have been using throughout our code.

4.3 Auxiliary functions

The following macros are used in several places of the nomogram code and are described first.

The first macro `DL` (defined with `def` and taking i as a parameter) transforms an integer i from 1 to 7 into a character from A to G and is used to display the dominical letter:

```
def DL(expr i)=char(64+i) enddef;
```

The macro `gn_epact` returns a pair made of the Golden Number and the Julian epact associated to the i-th point in column II (see figure 4), 1 being at the bottom. For $i = 2$, for instance, this macro returns $(7, 14)$. The macro is defined with `vardef`, which is a variant of `def` making it possible for the `G` and `JE` variables to have only a local scope after their `save` declaration.

```
vardef gn_epact(expr i)=
  save G,JE;
  G=1+((9-11i) mod 19);
  JE=(11G-3) mod 30;
  (G,JE) % value returned
enddef;
```

The macro `gn_epactl` returns the value of the Golden Number and the point in column IV associated to the i-th point in column II, 1 being at the bottom (see figure 4). For $i = 2$, for instance, this macro returns $(7, 4)$, because it is the 4th point of column IV which is associated to $G = 7$ (the second point being empty).

```
vardef gn_epactl(expr i)=
  save G,JE,JEL;
  G=1+((9-11i) mod 19);
  JE=(11G-3) mod 30;
  JEL=30-((JE+12) mod 30);
```

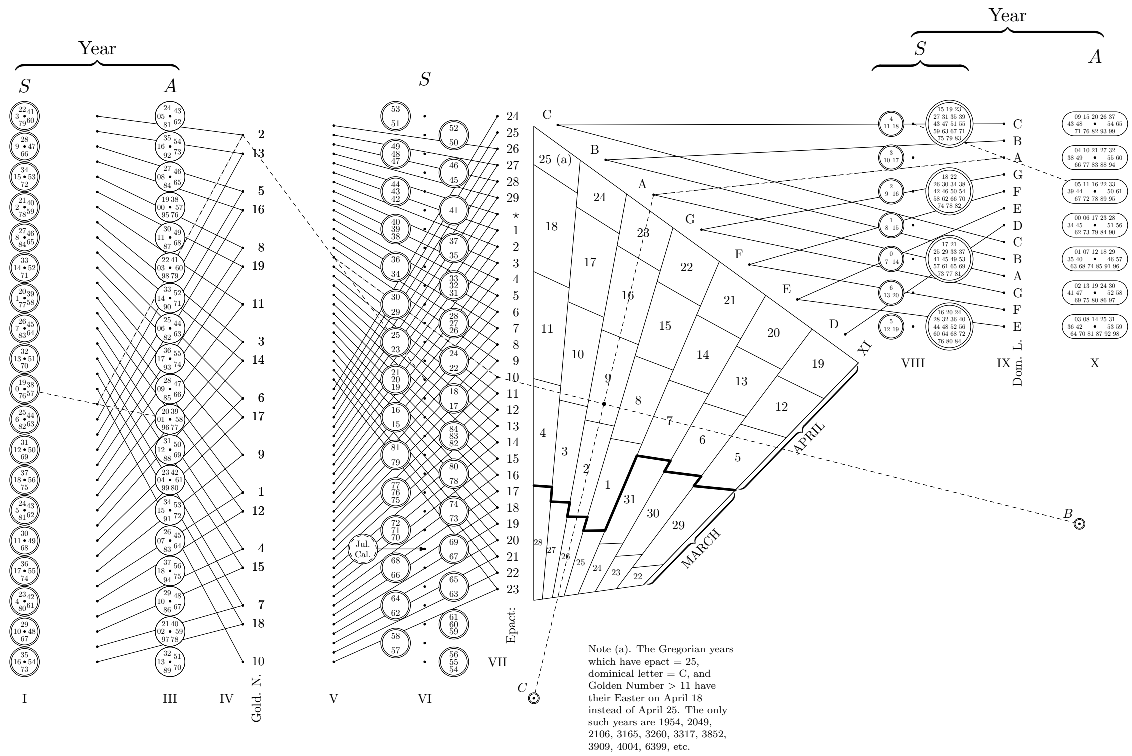

Figure 9: The complete METAPOST version of the nomogram.

```
  (G,JEL) % value returned
enddef;
```

`whatever` is a very useful METAPOST instruction which represents a yet undefined and *unnamed* scalar value. It is then possible to solve linear equations very easily, for instance finding point P such that $\overrightarrow{OP} = \overrightarrow{OA} + x\vec{u} = \overrightarrow{OB} + y\vec{v}$, x and y being the unknowns, with

```
  OP=OA+whatever*u=OB+whatever*v;
```

`OP`, `OA`, `OB`, `u`, and `v` being points (or complex values).

Interestingly, the two values of `whatever` (corresponding to the unknowns x and y) are usually not equal, which is why `whatever` should not be viewed as the name of a variable. In the previous example, we were more interested in the position of P than in the values of x and y, and P is merely the intersection of two lines.

In our code, we use the macro `whateverpair` which is the equivalent of `whatever` for pairs. It defines a "fresh" pair of numerical values (which need not be equal, despite the way they are defined, for the reason given above).

```
def whateverpair=
  (whatever,whatever)
enddef;
```

The next three macros are defined for formatting purposes. The first macro `ep_st` formats the epact value so that it fits on two characters, and the value 0 is displayed as '⋆'. `textext` is the main macro provided by the `latexmp` package.

```
def ep_st(expr i)=
  if i=0:
    textext("\phantom{0}$\star$")
  elseif i<10:
    textext("\phantom{0}"&decimal(i))
  else:
    textext(decimal(i))
  fi
enddef;
```

The second macro `gstring` is somewhat similar, but only formats a one or two-digit value with a two-digit width, forcing the value to have the same vertical size as an opening parenthesis, for alignment purposes.

```
def gstring(expr i)=
  if i<10:
    textext("\vphantom{(}\phantom{0}"
              &decimal(i))
  else:
    textext("\vphantom{(}"&decimal(i))
  fi
enddef;
```

The third macro `tddec` (two-digits decimal) formats a one or two-digit value as two digits, by possibly adding a 0 in front of it.

```
def tddec expr i=
  if i<10: "0" & decimal(i)
  else:
    decimal(i)
  fi
enddef;
```

4.4 Defining the points

In this section, we define the various points used in the construction of the nomogram.

4.4.1 Variables

For the points in the different columns, we mainly use two arrays of pairs:

```
pair col[][],col[]a[];
```

The points in column I will be stored in the variables `col[1][1]`, `col[1][2]`, `col[1][3]`, etc. The points in column II will be stored in `col[2][1]`, `col[2][2]`, `col[2][3]`, etc. In METAPOST, we can write `col1[1]` instead of `col[1][1]`, and this will simplify a little bit our code.

The second array (`col[]a[]`) is only used for the centers of the circles which are along column VI.

4.4.2 First points

The points in columns I to VII are set easily. The first and third columns have 19 points each, the second and fourth columns have 37 points each, column V has 58 points and columns VI and VII both have 30 points. All these points are linearly set and the points in columns II and V are obtained by bisecting segments linking points from adjacent columns.

The first three columns are straightforward to set, using a `height` constant defined elsewhere (and not described in this article):

```
for i:=1 upto 19:
  col1[i]=(0,(i-1)*height/18);
endfor;
for i:=1 upto 19:
  col3[i]=(40u,(i-1)*height/18);
endfor;
for i:=1 upto 37:
  col2[i]=
    ((xpart(col1[1])+xpart(col3[1]))/2,
     .5*(i-1)*height/18);
endfor;
```

Column IV is a bit more tricky, and for each of the 37 points in column II, the macro `gn_epactl` returns a pair made of the Golden Number associated to this point, and of the corresponding point in column IV. The Golden Number is not used here. The value of `col4` is then set:

```
for i:=1 upto 37:
  E:=ypart(gn_epactl(i));
  col4[E]=(60u,(E-1)*height/29);
endfor;
```

Columns V to VII are also easily set. In the case of column VI, additional points `col6a` are defined for the positions of the circles offset in that column. The points in column VII are shifted upwards by a certain amount (here $20u$, u being here equal to 1 mm).

```
for i:=1 upto 30:
  col6[i]=(110u,(i-1)*height/29);
endfor;
for i:=1 upto 58:
  col5[i]=
     ((xpart(col4[1])+xpart(col6[1]))/2,
      .5*(i-1)*height/29);
endfor;
for i:=1 upto 30:
  if i mod 2=0:
    col6a[i]=col6[i]-(8u,0);
  else:
    col6a[i]=col6[i]+(8u,0);
  fi;
endfor;
for i:=1 upto 30:
  col7[i]=(130u,
           20u+(i-1)*(height-20u)/29);
endfor;
```

4.4.3 Easter table

The whole Easter table is obtained by setting points B and C, as well as four corners of the table. B and C can be positioned freely.

```
vardef define_easter_table=
  save corner,p;pair corner[];
  C=(xpart(col7[1])+10u,-10u);
  B=(xpart(C)+150u,ypart(col7[5]));
```

We define two additional points in column VII, one above the 30th (`col7[31]`), and one below the first (`col7[0]`):

```
  col7[1]-col7[0]=col7[31]-col7[30]
                 =col7[2]-col7[1];
```

The shape of area XI is defined by its four corners:

```
  corner1=whatever[col7[0],B]
         =C+whatever*up;
  corner3=.3[B,col7[31]];
  corner2=(C--corner3) intersectionpoint
          (B--corner1);
  corner4=whatever[B,col7[31]]
         =C+whatever*up;
```

Then, the whole area determined by the points `corner1`, `corner2`, `corner3`, and `corner4` is divided into eight slices, only seven of which will be drawn (figure 10). The first slice contains the Easter dates March 28, April 4, 11, 18 and 25. The second slice contains the Easter dates March 27, April 3, 10, 17, and 24, and so on. The eighth slice is not drawn, but defined for practical reasons. These eight slices are limited by nine boundary lines. The first seven slices correspond to the dominical letters C, B, A, G, F, E, and D shown in column XI:

The slices are represented using two two-dimensional arrays, `s[]l[]` and `s[]r[]`. The Easter area is divided into slots, and each slot is a quadrilateral. Two of the vertices of each quadrilateral are located on one slice boundary, and the two others are located on another boundary. If we now consider a boundary between slices, which is a (more or less) vertical segment, this boundary contains points from which some segments go to the left, and other go to the right (more or less). In the former case, the points are given by the array `s[]l[]` ('l' for left), and in the latter case by the array `s[]r[]` ('r' for right). All the boundaries contain 10 points. The points of the second boundary, for instance, are `s2l0`, `s2l1`, `s2l2`, `s2l3`, `s2l4`, `s2l5`, `s2r0`, `s2r1`, `s2r2`, `s2r3`, `s2r4`, `s2r5`. `s2l0` is equal to `s2r0`, and `s2l5` to `s2r5`.

The first part of the code defines the beginnings and ends of each boundary line:

```
for i=1 upto 9:
  s[i]l0=s[i]r0
        =(((i-1)/8))[corner4,corner3];
  s[i]l5=s[i]r5
        =whatever[corner1,corner2]
        =whatever[s[i]l0,C];
endfor;
```

We then divide each of the eight boundaries four times. i is the boundary number and goes from left to right. Eight vertical lines enclose the 35 Easter slots. j varies over the horizontal inner divisions. A division is made so that the line going through B and the division falls exactly between two epact values in column VII:

```
for i=1 upto 8:
  for j=1 upto 4:
    p:=30-i-(j-1)*7;
    if i<8:
      % division leaving to the right
      %  of vertical line i
      s[i]r[j]=(s[i]l0--s[i]l5)
                 intersectionpoint
            (B--.5[col7[p],col7[p-1]]);
    fi;
    if i>1:
      % division leaving to the left
      % of vertical line i
      s[i]l[j]=(s[i]l0--s[i]l5)
                 intersectionpoint
            (B--.5[col7[p+1],col7[p]]);
      fi;
    endfor;
  endfor;
enddef;
```

Finally, the points in column XI are obtained from the upper boundary of the Easter area. They are put on a line parallel to (`corner3`, `corner4`) and in the middle of the slices (as seen from C).

```
vardef define_dominical_letters=
  save shift;pair shift;
  shift=(3u,3u);
  for i=1 upto 8:
    col11[i]
      =whatever[C,.5[s[i]r0,s[i+1]l0]]
      =whatever[s1r0+shift,s8l0+shift];
  endfor;
enddef;
```

4.4.4 Last points

The points in columns VIII to X are determined as follows:

```
  for i=1 upto 7:
    col8[i]=s8l0
      +(15u,10u+(i-1)*ypart(s1l0-s8l0)/7);
    col10[i]=col8[i]+(50u,0);
  endfor;
  for i=1 upto 13:
    col9[i]=(xpart(col8[1]+col10[1])/2,
             ypart(col8[1])
                 +(i-1)*(ypart(col8[7]
                          -col8[1]))/12);
  endfor;
```

4.5 Drawing the connections

Connections between columns II and IV are drawn by the following code:

```
  for i:=1 upto 37:
    draw col2[i]
          --col4[ypart(gn_epactl(i))];
  endfor;
```

Connections between columns V and VII are drawn by the following code:

```
  for i:=1 upto 58:
    draw col5[i]--col7[1+((i-1) mod 30)];
  endfor;
```

Connections between columns IX and XI are obtained by the following code:

```
  for i=1 upto 13:
    draw col9[i]--col11[1+(13-i) mod 7];
  endfor;
```

4.6 Drawing the circles

Double circles are drawn using a straightforward macro not described here. For column I, the circles are drawn with:

```
  for i:=1 upto 19:
```

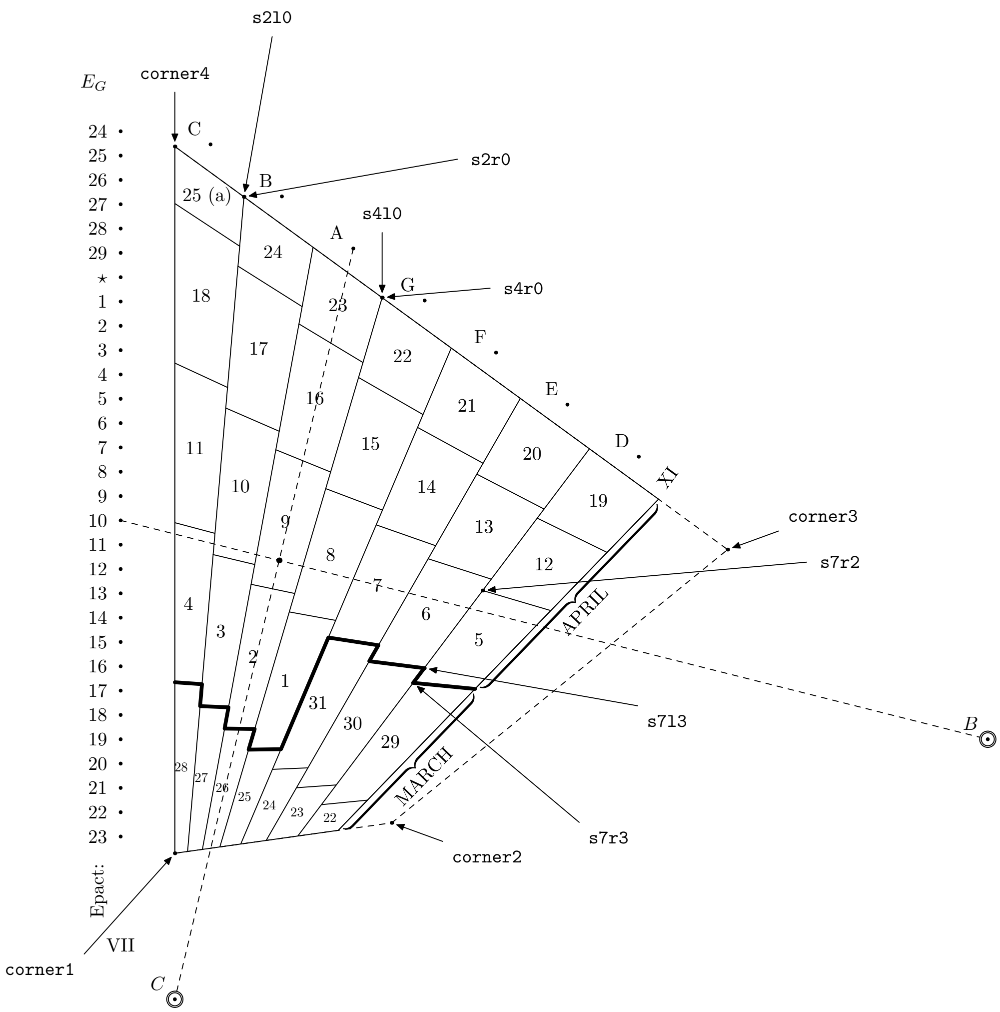

Figure 10: The Easter area, for the determination of Easter using the dominical letter and the mean Gregorian epact. This table shows one exception (note (a)), corresponding to the case $\overline{E_G} = 25$, $G > 11$ and $DL = C$. The cases $\overline{E_G} = 24$ and $(\overline{E_G} = 25) \wedge (G > 11)$ are gathered in the table. The figure shows that $DL = A$ and $\overline{E_G} = 10$ puts Easter on April 9.

```
    draw_dbl_circle(.9diam1,diam1,col1[i]);
  endfor;
```

The circles in the other columns are obtained similarly.

4.7 Labeling the points

For the labels of columns I, III and VI, we first build special strings which will be used later to typeset the labels. These strings are stored in the following variables:

```
string col[][]st;
```

4.7.1 Preparing the labels

Labels in the first column are defined as follows. We go through every secular part from 0 to 84 and find the position to which it belongs, using the formulæ found earlier. There are two cases, either the string was not yet defined (in which case `unknown col1[p]st` is true and we assign its first value, or it was already defined, and we append a new value with a comma in between. The comma will be useful later, when the string is analyzed.

```
vardef define_col_one_labels=
  save p;
  for i=0 upto 84:
    p:=1+(9+3i) mod 19;
    if unknown col1[p]st:
      col1[p]st=decimal(i);
    else:
      col1[p]st
        :=col1[p]st & "," & decimal(i);
```

```
    fi;
  endfor;
enddef;
```

Labels in the third and sixth columns are defined similarly:

```
vardef define_col_three_labels=
  save p;
  for i=0 upto 99:
    p:=1+(15+12i) mod 19;
    if unknown col3[p]st:
      col3[p]st=decimal(i);
    else:
      col3[p]st
        :=col3[p]st & "," & decimal(i);
    fi;
  endfor;
enddef;
```

In the sixth column, we use the value of the Gauss constant M and the computation is only done for values of the secular part between 15 and 84, since earlier centuries lead to a constant value of M.

```
vardef define_col_six_labels=
  save p,M;
  for i=15 upto 84:
    M:=(15+i-floor(i/4)
          -floor((8i+13)/25)) mod 30;
    p:=1+(M+21) mod 30;
    if unknown col6[p]st:
      col6[p]st=decimal(i);
    else:
      col6[p]st
        :=col6[p]st & "," & decimal(i);
    fi;
  endfor;
enddef;
```

4.7.2 Column I

Once the strings for the labels have been defined, these strings can be processed and the labels can be drawn. The macro processing the labels in columns I and III is `col_one_three_f`. This macro, as well as `col_six_f`, first counts the number of elements in the `list` parameter and stores it in `n`. It does so by analyzing the comma-separated string `list` with `scantokens`, which evaluates a string as if it were normal METAPOST code.

```
vardef col_one_three_f(expr list,l,c)=
  save n,i;n=0;
  for $=scantokens(list):
    n:=n+1;
  endfor;
  i=0;
  for $=scantokens(list):
    i:=i+1;
    label(textext(if c=3: (tddec $)
              else: decimal $ fi)
                 scaled .6,
               col[c][l]
                  +((if c=3: 2.5u
                     else: 2u
                     fi,0)
                      rotated
                        (180-(i-1)*360/n)));
  endfor;
enddef;
```

Now, the labels are drawn with:

```
  for i=1 upto 19:
    col_one_three_f(col1[i]st,i,1);
  endfor;
  label(textext("I"),col1[1]-(0,10u));
  label(textext("$S$") scaled 1.5,
        col1[19]+(col1[19]-col1[18]));
```

4.7.3 Column III

The labels of column III are drawn using the same macro as for column I:

```
  for i=1 upto 19:
    col_one_three_f(col3[i]st,i,3);
  endfor;
  label(textext("III"),col3[1]-(0,10u));
  label(textext("$A$") scaled 1.5,
        col3[19]+(col3[19]-col3[18]));
```

4.7.4 Column IV

The labels in column IV are drawn using the macro `gn_epactl` seen above.

It should be noted that some of the values here are written twice, but this causes no harm.

```
  pair GNE;
  for i:=1 upto 37:
    GNE:=whateverpair;
    GNE=gn_epactl(i);
    label.rt(gstring(xpart(GNE)),
          col4[ypart(GNE)]);
  endfor;
```

4.7.5 Column VI

The labels in the circles of column VI are drawn by processing the strings `col6[]st` which were prepared above. The postprocessing is done using the macro `col_six_f`:

```
vardef col_six_f(expr list,l)=
  save n,i;n=0;
  for $=scantokens(list):
    n:=n+1;
  endfor;
  i=0;
  for $=scantokens(list):
    i:=i+1;
    if n>1:
```

If there is more than one value, the extreme values are put at $2u$ below and above the center, and the other values (if any) are spread evenly in-between:

```
      label(textext(decimal $) scaled .7,
            col6a[l]
             +(0,-2u+(i-1)*(4u/(n-1))));
    else:
```

If there is only one value, it is centered; there is only one such case:

```
      label(textext(decimal $) scaled .7,
            col6a[l]);
    fi;
  endfor;
enddef;
```

The labels are then drawn with:

```
  for i=1 upto 30:
    col_six_f(col6[i]st,i);
  endfor;
  label(textext("VI"),col6[1]-(0,10u));
  label(textext("$S$") scaled 1.5,
        col6[30]+2(col6[30]-col6[29]));
```

4.7.6 Column VII

In this column, we merely output the value of the epact.

```
  for i:=1 upto 30:
    label.rt(ep_st((24-i) mod 30),col7[i]);
  endfor;
  label(textext("VII"),col7[1]-(0,20u));
  label.rt(textext("Epact:") rotated 90,
           col7[1]-(0,10u));
```

4.7.7 Column VIII

In this column, there are labels for the Julian calendar on the left, and labels for the Gregorian calendar on the right. In the first case, there are always three values of S in each circle, and the labels can be produced by a simple loop.

```
  for i=1 upto 7:
    for j=1 upto 3:
      v:=((4+i) mod 7)+(j-1)*7;
      label(textext(decimal(v)) scaled .5,
            col8[i]-(col_shift_eight_a,0)
             +((0,1.4u)
                 rotated ((j-1)*120)));
    endfor;
  endfor;
```

For the Gregorian calendar, there are some irregularities, and we have decided to explicit each line of the labels. The following lines could be parameterized, but it's not worth it.

```
  secular_year(1,1)(16,20,24);
  secular_year(1,2)(28,32,36,40);
  secular_year(1,3)(44,48,52,56);
  secular_year(1,4)(60,64,68,72);
  secular_year(1,5)(76,80,84);
  secular_year(2,1)(17,21);
  secular_year(2,2)(25,29,33,37);
  secular_year(2,3)(41,45,49,53);
  secular_year(2,4)(57,61,65,69);
  secular_year(2,5)(73,77,81);
  ...
```

The macro `secular_year` is defined as follows. It distributes all lines evenly, since there are always five of them:

```
vardef secular_year(expr i,j)(text sec)=
  save vd;
  % vertical shift of the first line
  vd=4u;
  label(textext(sval(sec)(decimal))
           scaled .5,
        col8[2i-1]+(10u,vd-(j-1)*.5vd));
enddef;
```

The `sval` macro builds a string with space-separated values:

```
vardef sval(text sec)(text f)=
  save s;string s;
  for $=sec:
    if unknown s:
      s=f $;
    else:
      s:=s & " " & f $;
    fi;
  endfor;
  s
enddef;
```

4.7.8 Column IX

In this column, we show all dominical letters resulting from the combination of the two parts of the year.

```
  for i=1 upto 13:
    label.rt(textext(DL(1+(3+i) mod 7)),
             col9[i]);
  endfor;
  label(textext("IX"),col9[1]-(0,10u));
  label.rt(textext("Dom. L.") rotated 90,
           col9[1]-(0,10u));
```

4.7.9 Column X

The labels in column X fit in rounded rectangles. In order to produce these "rectangles", we use the `rboxes` package and draw a rectangular box with rounded corners. There are seven boxes, `rb1` to `rb7`:

```
  rbr=rbox_radius;
  rbox_radius:=15pt;
  for i=1 upto 7:
    rboxit.rb[i]("");
    rb[i].c=col10[i];
    rb[i].dx=9u;rb[i].dy=3.3u;
    unfill bpath(rb[i]);
    drawboxes(rb[i]);
```

```
  endfor;
  rbox_radius:=rbr;
```

Once the boxes are drawn, their contents can be added. Each box has three lines, the upper and lower ones extend on the whole width, and the middle one is split in two parts, one left of the center, and the other one right of the center.

The upper and lower lines are produced with the `yn` (year number) macro, whose first parameter is the point number, and whose second parameter is the line number within the label, the first line being here at the top. The middle line is produced with `yn_left` and `yn_right`.

```
  yn(1,1)(3,8,14,25,31);
  yn_left(1)(36,42);
  yn_right(1)(53,59);
  yn(1,3)(64,70,81,87,92,98);
  ...
```

Then, we have the three macros for setting a year.

`yn` puts the labels at symmetric positions in the two cases in which it is called.

```
vardef yn(expr i,j)(text y)=
  save vd;
  % vertical shift of the first line
  vd=2u;
  label(textext(sval(y)(tddec)) scaled .5,
        col10[i]+(0,vd-(j-1)*vd));
enddef;
```

`yn_left` and `yn_right` just put the label at symmetric positions on the left and right of the central point selected by i:

```
def yn_left(expr i)(text y)=
  label.rt(textext(sval(y)(tddec)) scaled .5,
           col10[i]+(-10u,0));
enddef;

def yn_right(expr i)(text y)=
  label.lft(
     textext(sval(y)(tddec)) scaled .5,
     col10[i]+(10u,0));
enddef;
```

4.7.10 Column XI

In this column, we output the seven dominical letters.

```
  for i=1 upto 7:
    label.ulft(textext(DL(1+(10-i) mod 7)),
               col11[i]);
  endfor;
  label(textext("XI")
          rotated (angle(s1r0-s8l0)-90),
        .6[col11[8],col11[7]]);
```

4.8 Drawing the Easter grid

Once the various slices of the Easter grid have been defined, the grid can be drawn easily. We first draw the slots, then the labels.

4.8.1 The slots

The Easter slots are drawn with the following macro:

```
vardef draw_easter_table_slices=
  save oldpen;
  oldpen=savepen;
  % divisions between slices:
  for i=1 upto 8:
    draw s[i]l0--s[i]l5;
  endfor;
  % external boundary:
  draw s8l5--s8l0--s1l0--s1l5--cycle;
  % internal divisions:
  for i=1 upto 7:
    for j=1 upto 4:
      draw s[i]r[j]--s[i+1]l[j];
    endfor;
  endfor;
  % March/April divisions:
  pickup pencircle scaled 2pt;
  draw s8l3--s7r3--s7l3--s6r3--s6l3--
       s5r3--s5l4--s4r4--s4l4--s3r4--
       s3l4--s2r4--s2l4--s1r4;
  pickup oldpen;
enddef;
```

4.8.2 Easter grid labels

For the labels inside the Easter grid, we first define an auxiliary macro. This macro takes a slice number x and a position y within the slice, and puts the label *lab* in the middle of the corresponding slot:

```
def label_easter_slot(expr x,y,lab)=
  label(lab,.5[s[x]r[y],s[x+1]l[y+1]]);
enddef;
```

Now, the main macro filling the Easter grid slots is the following. We first fill every slot with the appropriate number, and add a special case for April 25th ($\overline{E_G} = 25$, $G > 11$ and $DL = C$):

```
vardef draw_easter_table_labels=
  save laban,march,april,note,sl,j;
  string march,april,note;
  % 35 dates from March 22 till April 25
  for i=1 upto 35:
    sl:=1+(7-(i mod 7)) mod 7;
    j:=4-floor((i-1)/7);
    label_easter_slot(sl,j,
       textext(if i=35:"25 (a)"
               else:
                 decimal(if i>10:
                           i-10
                         else:
                           i+21
```

```
                        fi)
              fi)
       if i<8:
         scaled .7
       fi);
  endfor;
```

Then, we need to add two braces, as well as the footnote. This is done as follows:

```
  laban=angle(s810-s815);
  march="$\underbrace{\kern" &
     decimal(arclength(s813--s815)-5) &
        "bp}_{\hbox{MARCH}}$";
  april="$\underbrace{\kern" &
     decimal(arclength(s810--s813)-5) &
        "bp}_{\hbox{APRIL}}$";
  note="\footnotesize "&
         "\parbox{4cm}{\raggedright "&
         "Note (a)...}";
  label(textext(march) rotated laban,
        .5[s813,s815]
        +3u*unitvector((s810-s815)
                        rotated -90));
  label(textext(april) rotated laban,
        .5[s810,s813]
         +3u*unitvector((s810-s815)
                         rotated -90));
  label(textext(note),C+35u*right);
enddef;
```

5 Conclusion

We have eventually completed the analysis and reconstruction of Garrigues's nomogram. To some extent, the reconstruction was straightforward, and could have been achieved without a deep understanding of the nomogram, only by a mere observation. However, a good reconstruction almost always benefits from an initial analysis, and is useful if the structure has to be explained. Such conclusions had already been made in a previous work on a complex drawing in descriptive geometry (Roegel, 2007).

6 Acknowledgements

It is a pleasure to thank Damien Wyart who, many years ago, drew our attention to Garrigues' article and led us to redraw this nomogram.

References

Bien, Reinhold. "Gauß and Beyond: The Making of Easter Algorithms ". *Archive for history of exact sciences* **58**(5), 439–452, 2004.

Dershowitz, Nachum, and E. M. Reingold. *Calendrical calculations*. Cambridge: Cambridge University Press, 2008. 3rd edition.

d'Ocagne, Maurice. *Traité de nomographie : Étude générale de la représentation graphique cotée des équations à un nombre quelconque de variables, applications pratiques*. Paris: Gauthier-Villars, 1921. 2nd edition (first edition in 1899).

Garrigues, Damien. "Généralisation de la formule pascale de Gauss: Nomogramme du Calendrier perpétuel". *Annales françaises de chronométrie* pages 47–60, 1939.

Gauss, Karl Friedrich. "Berechnung des Osterfestes". *Monatliche Correspondenz zur Beförderung der Erd- und Himmels-Kunde, herausgegeben vom Freiherrn Franz Xavier G. von Zach* **2**, 121–130, 1800. In: (Gauss, 1973), volume 6, pages 73–79.

Gauss, Karl Friedrich. "Berichtigung zu dem Aufsatze: Berechnung des Osterfestes. Mon. Corr. 1800 Aug, S. 121". *Zeitschrift für Astronomie und verwandte Wissenschaften, herausgegeben von B. von Lindenau und J. G. F. Bohnenberger* **1**, 158, 1816. Correction of (Gauss, 1800). In: (Gauss, 1973), volume 11/1, page 201. Notes by Alfred Loewy.

Gauss, Karl Friedrich. *Werke*. New York: G. Olms, 1973. Reprint of the 1863–1933 edition.

Goossens, Michel, F. Mittelbach, S. Rahtz, D. Roegel, and H. Voss. *The LaTeX Graphics Companion, Second Edition*. Boston: Addison-Wesley, 2008.

Hobby, John. "METAPOST: A User's Manual". 2008. Updated version of the original manual; available at `http://tug.org/docs/metapost/mpman.pdf`.

Knuth, Donald Ervin. *Fundamental Algorithms, volume 1 of The Art of Computer Programming*. Reading, Massachusetts: Addison-Wesley, 1997.

Roegel, Denis. "The missing new moon of A.D. 16399 and other anomalies of the Gregorian calendar". 2004. Unpublished.

Roegel, Denis. "A complex drawing in descriptive geometry". *TUGboat* **28**(2), 218–228, 2007.

◇ Denis Roegel
LORIA — BP 239
54506 Vandœuvre-lès-Nancy cedex, France
roegel (at) loria dot fr
http://www.loria.fr/~roegel